PAPER • **OPEN ACCESS**

WEISS: Wasserstein efficient sampling strategy for LLMs in drug design

IOP Publishing Mach. Learn.: Sci. Technol. 6 (2025) 025048 https://doi.org/10.1088/2632-2153/addc33

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
12 December 2024

REVISED
10 May 2025

ACCEPTED FOR PUBLICATION
22 May 2025

PUBLISHED
3 June 2025

PAPER

WEISS: Wasserstein efficient sampling strategy for LLMs in drug design

Riccardo Tedoldi[1,2,*], **Junyong Li**[1,3], **Ola Engkvist**[1,3], **Andrea Passerini**[2], **Annie M Westerlund**[1] and **Alessandro Tibo**[1,*]

[1] Molecular AI, Discovery Sciences, R&D AstraZeneca, Gothenburg, Sweden
[2] Department of Information Engineering and Computer Science, University of Trento, Trento, Italy
[3] Data Science and AI, Computer Science and Engineering,Chalmers University of Technology, Gothenburg, Sweden
[*] Authors to whom any correspondence should be addressed.

E-mail: riccardo.tedoldi@astrazeneca.com and alessandro.tibo@astrazeneca.com

Keywords: drug design, autoencoders, efficient sampling, LLMs

Supplementary material for this article is available online

Abstract

Autoregressive models have gained popularity in the field of drug design due to their capability to sample novel molecules from a vast chemical space efficiently. Sampling novel and diverse molecules in an efficient manner is a crucial aspect, as it is important for downstream tasks such as reinforcement learning to identify novel molecules with pre-defined desired properties. Existing sampling strategies like multinomial sampling and beam search often struggle with mode collapses or are computational inefficient, respectively. To address these limitations, we introduce WEISS (Wasserstein efficient sampling strategy), a framework that seamlessly enables autoregressive models to efficiently sample diverse molecules. Our approach, which draws inspiration from the Wasserstein autoencoder, is compatible with any encoder–decoder-based autoregressive model. We show that WEISS effectively mitigates mode collapsing while maintaining token sampling speed 25 times faster than beam search. Secondly, we showcase the efficacy of the proposed method for various drug design tasks such as molecular property optimization and single-step retrosynthesis prediction.

1. Introduction

Small molecules represent the primary source of pharmaceutical drugs. Designing optimal molecules is a major challenge in drug discovery. De novo drug design aims to discover novel molecules with optimized physical and chemical properties, such as high efficacy and low toxicity. Furthermore, discovering new potential drugs is not sufficient on its own; synthesis planning (how to make them) is another important task in drug design. Advancements in deep learning [1] and natural language processing (NLP) [2] have had significant impact in the field of de novo drug design. These include lead discovery [3], lead optimization [4] and single-step [5] and multi-step [6] retrosynthesis prediction, target-specific drug design [7], and multi-objective drug optimization [8].

There are two main approaches to address drug design problems: (1) training generative models that suggest novel molecules with the desired properties [4], and (2) pre-training unconditional models on large datasets and then steering them towards chemical regions of interest using reinforcement learning (RL) [9]. RL operates by sampling a large number of molecules from the pre-trained model and then adjusting its weights based on a scoring function which rewards the generated compounds that exhibit desirable properties. This second approach offers the advantage of greater flexibility in property optimization while adapting to the diverse needs of different projects. Desired properties may, for example, vary based on the target localization, as central nervous system drugs often require stricter constraints on lipophilicity and molecular weight [10], or based on the route of administration, with oral drugs typically having a lower molecular weight and fewer hydrogen bonds compared to intravenously administered drugs [11]. To ensure

the effectiveness of the RL approach, however, it is essential that the pre-trained model generates samples efficiently.

In this context, common choices for generative pre-trained models are autoregressive encoder–decoder based models such as seq2seq long short term memory (LSTM) [12] or large language models trained on SMILES [13], a string representation for encoding molecules. While some might argue that representing molecules as graphs would be a more natural choice, studies have demonstrated that models trained using SMILES representations have similar performance [14] in terms of validity, uniqueness and novelty [15]. In addition, SMILES representations are cheaper and simpler to use and handle than molecular graphs when considering data processing and inference.

SMILES-based models can use multinomial sampling and its variants (such as top-k and top-p sampling) or beam search to generate multiple molecules. However, multinomial sampling in the context of drug design suffers from mode collapse (sampling the same output sequence multiple times). Tuning the temperature parameter can mitigate this phenomenon [16], but multinomial sampling may still suffer from poor diversity. In contrast, beam search avoids the duplicates issue by design. However, it is computationally expensive because it simultaneously maintains and re-evaluates the probability of multiple candidate sequences (so-called beams) several times. This process to sample the optimal sequences (the most likely) requires scoring and ranking an exponentially growing number of sequence continuations, leading to significant computational overhead as the search space expands. Though beam search expands always a subset of those to deal with this exponentially growing number of sequence continuation not guaranteeing its search optimality. Even though Beam Search is a heuristic, its characteristics make it effective in, for instance, RL or single-step retrosynthesis settings. However, the computational cost to sample multiple sequences makes it impractical in applications where a large beam size is needed, and the model is called iteratively.

To overcome the limitations of existing solutions, in this paper, we introduce WEISS (Wasserstein efficient sampling strategy), a new approach for molecule generation based on Wasserstein autoencoders (WAEs) [17]. WEISS is designed to sample from autoregressive encoder–decoder models, like transformers [1], effectively exploiting the efficiency of multinomial sampling with a solution to the issue of mode collapse. We demonstrate the advantages of WEISS in multiple areas. Firstly, in a molecular optimization task, where we train a transformer model using our framework and then employ RL to optimize specific physical and chemical properties. Secondly, in a retrosynthesis task, where we train a transformer to predict feasible and diverse reactions for making a target molecule. Despite WEISS's initial development for molecular research, its applicability extends beyond this field. To highlight this versatility, we also present results for NLP tasks.

We summarize the key contributions as follows:

- We introduce WEISS, a novel efficient and versatile framework for encoder–decoder autoregressive models;
- We demonstrate the model's capabilities in unconditional and conditional molecular generation, demonstrating high validity, diversity, and novelty compared to multinomial sampling with state-of-the-art frameworks;
- We assess the method's applicability for single-step retrosynthesis prediction using a fine-tuned transformer, showing significant diversity improvements;
- We demonstrate the generality of WEISS by showing promising results on NLP question answering tasks as part of the supplementary material.

The code is available at https://github.com/r1cc4r2o/weiss [18].

2. Methods

We aim to design a model that maps sequences into other sequences. Let $\mathcal{S}$ be a sequence space, where $x = x_{0:T-1} = (x_0, \ldots, x_{T-1})$, $x \in \mathcal{S}$, is a sequence of length T. Each element or token x_i of x is represented by an integer taken from a vocabulary $V = \{0, \ldots, |V| - 1\}$. Finally, we consider a dataset $\mathcal{D} = \{(x, y) \mid x, y \in \mathcal{S} \times \mathcal{S}\}$, where x and y belong to the same space $\mathcal{S}$.

Our goal is to create a model for the conditional probability distribution $p(y|x)$ that enables conditional sampling of sequences by incremental token sampling, that is

$$p(y|x) = p(y_0|x) \prod_{t=1}^{U-1} p(y_t|y_{t-1}, \ldots, y_0, x) \tag{1}$$

where U is the length of y. We consider a learnable autoregressive model $f_\theta : \mathcal{S} \times \mathcal{S} \to [0,1]^{|V|}$ parametrized by θ as a proxy for the probability distribution in equation (1). f_θ can be, e.g. an LSTM [19] or a transformer-based model as [1] and [20].

The autoregressive model f_θ consists of two components: an encoder $\Phi_E : \mathcal{S} \to \mathbb{R}^{T\times N}$ and a decoder $\Phi_D : \mathcal{S} \times \mathbb{R}^{T\times N} \to [0,1]^{|V|}$. The encoder takes an input sequence $x \in \mathcal{S}$ and converts it into a higher-dimensional representation $\Phi_E(x) \in \mathbb{R}^{T\times N}$. Given a pair $(x,y) \in \mathcal{D}$, at training time, the decoder takes as input $y_{0:t}$ and $\Phi_E(x)$, and produces the probabilities of the next token $\hat{y}_{t+1}$.

Finally, f_θ is trained by minimizing the negative log-likelihood (NLL)

$$L_{\text{NLL}}(x,y) = -\sum_{t=1}^{U} \log f_\theta\left(x, y_{0:t-1}\right)[y_t], \tag{2}$$

over the training set $\mathcal{D}$. During inference, the decoder predicts one token at a time. We start with an initial token and then feed the model with the previously generated tokens along with the input sequence x, that is $\hat{y}_{t+1} \sim \Phi_D(\hat{y}_{0:t}, \Phi_E(x))$.

Multiple strategies exist for sampling from f_θ, including but not limited to multinomial sampling, top-p, top-k, and beam search. However, when standard autoregressive models use highly efficient techniques like multinomial sampling, top-p, and top-k, they often collapse into generating the same output [16]. Conversely, strategies such as beam search and its variants effectively mitigate the mode collapsing [21], by systematically sampling diverse combinations of tokens. However, they involve substantial memory and computational costs due to the need of enumerating the top-B best sequences, where B represents the beam size.

In the following sections, we will introduce our proposed strategy that preserves the efficiency of multinomial sampling, top-p, top-k, while mitigating the mode collapse effect. Our strategy encourages output variation without incurring a high computational overhead typical of beam search, thereby achieving a favorable balance between diversity and efficiency.

2.1. Sampling through latent variables

Our goal is to capture and represent the correlation between x and y through a latent variable z. By sampling different values of z, we allow the model to produce multiple outputs given a single sequence as input. Variational autoencoder (VAE) [22] and WAE [17] are two well-known approaches to enforce z to follow a distribution. Since WAE has been shown to produce higher-quality samples and providing more robust representations [17], we selected this approach over VAE.

2.1.1. Learning the latent variable

We now introduce a framework that expands the expressiveness of f_θ by including a continuous latent variable $z \in \mathbb{R}^d$ into equation (1), so that y depends on x and z, that is $p(y|x,z)$. By applying the Bayesian rule we obtain

$$p(z|x,y) = \frac{p(y|x,z)\,p(x,z)}{\int_z p(y|x,z)\,p(x,z)\,\mathrm{d}z}, \tag{3}$$

which is intractable as the integral at the denominator does not have a closed-from solution in general. However, similarly to WAE, we regularize $p(z|x,y)$ so that $p(z) = \int p(z|x,y)\mathrm{d}x\mathrm{d}y$ matches a Gaussian distribution $q(z) = \mathcal{N}(\underline{0}, I)$, where $\underline{0} \in \mathbb{R}^d$ is a vector containing all zero entries, and I is the identity matrix. To enforce the match between p and q we used the maximum mean discrepancy (MMD), defined as

$$\text{MMD}(p,q) = \frac{1}{n(n-1)} \sum_{i\neq j} k\left(z_i, z_j\right) + k\left(\hat{z}_i, \hat{z}_j\right) - \frac{2}{n^2} \sum_{i,j} k\left(z_i, \hat{z}_j\right), \tag{4}$$

where $z_0, \ldots, z_{n-1} \sim p(z|x,y)$, $\hat{z}_0, \ldots, \hat{z}_{n-1} \sim q(z)$, and k a positive-definite reproducing kernel.

We model $p(z|x,y)$ by first obtaining the feature representations $g_x = \Phi_E(x)$ and $g_y = \Phi_E(y)$. Note that Φ_E is used for both x and y at training time. These representations are then aggregated using a function $\Xi : \mathbb{R}^{|\mathcal{S}|\times N} \to \mathbb{R}^N$. Ξ (the mean in our experiments) converts a sequence of feature vectors into a single vector, thereby summarizing the entire sequence as $h_x = \Xi(g_x)$ and $h_y = \Xi(g_y)$. Finally, an MLP $\Psi_E : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^d$, with $d << n$, maps the summaries of x and y into the latent variable z, that is $z = \Psi_E(h_x, h_y)$. z needs to have a small dimension in order to prevent the curse of dimensionality with the MMD. On the other hand, we also need a representation of z that is aligned with the dimension of the decoder input. We achieve this via another MLP $\Psi_D : \mathbb{R}^d \to \mathbb{R}^N$, that we use to model samples from $p(h_x + h_y|z)$. Unlike in an autoencoder, we do not have direct supervision over $p(h_x + h_y|z)$, so as shown

below, it needs to be regularized differently. Note that the subsequent analysis remains valid even if one chooses to use for e.g. $p(h_x|z)$ and $p(h_y|z)$ instead of $p(h_x + h_y|z)$.
The full architecture can be synthesized, at training time, as follows

$$\begin{aligned} &g_x = \Phi_E(x), \quad g_y = \Phi_E(y), \quad h_x = \Xi(g_x), \quad h_y = \Xi(g_y) \\ &z = \Psi_E(h_x, h_y) \\ &p(\hat{y}_t|y_{0:t-1}, x, z) = \Phi_D(y_{0:t-1}, g_x + \Psi_D(z)). \end{aligned} \tag{5}$$

Here $\Psi_D(z)$ represents a coarse-grained representation of the input–output pair, while g_x contains a fine-grained representation of the input. We combine the two to obtain a fine-grained representation of the input–output pair to be fed to the decoder.

2.1.2. Bias correction
Modelling z as described in the previous section introduces a bias. This bias is introduced in equation (5) where the term $g_x + \Psi_D(z)$ yields a double contribution from the input sequence x. Since $\Psi_D(z)$ models $p(h_x + h_y|z)$, its expected value is

$$\mathbb{E}[h_x + h_y|z] = \mathbb{E}[h_x|z] + \mathbb{E}[h_y|z].$$

As such, we are adding the redundant contribution of $\mathbb{E}[h_x|z]$ to g_x. In the following, we show how to compensate for this bias by including an additional term in the loss function which allows us to subtract the bias directly from $g_x + \Psi_D$. Let $z_0 \sim \mathcal{N}(\underline{0}, \sigma I)$ be sample close to 0, with small variance σ, and let us consider

$$\begin{aligned} \mathbb{E}[h_x|z_0] &= \sum_x h_x\, p(h_x|z_0) = \sum_{x,y} h_x\, p(h_x, h_y|z_0) \\ &= \sum_{x,y} h_x\, p(z_0|h_x, h_y)\, p(h_y|h_x)\, p(h_x)/p(z_0). \end{aligned} \tag{6}$$

Assuming $p(h_y|h_x) = 0$ if and only if $h_x \neq h_y$, equation (6) becomes

$$\mathbb{E}[h_x|z_0] = \sum_x h_x\, p(z_0|h_x, h_x)\, p(h_x)/p(z_0). \tag{7}$$

Finally, we ensure that $p(z|h_x, h_x)$ is concentrated around z_0 by regularizing the training loss to impose $\mathbb{E}[z|h_x, h_x] = z_0$. Consequently, $p(z_0|h_x, h_x) \approx 1$, allowing equation (7) to be approximated as follows:

$$\mathbb{E}[h_x|z_0] \approx \sum_x h_x\, p(h_x)/p(z_0) = \mathbb{E}[h_x]/p(z_0). \tag{8}$$

The same results can be obtained for $\mathbb{E}[h_y|z_0]$. Treating $g_x = \mathbb{E}[h_x]/T$ and allowing λ to absorb the normalization terms, we can regularize the model architecture as follows

$$\begin{aligned} &g_x = \Phi_E(x), \quad g_y = \Phi_E(y), \quad h_x = \Xi(g_x), \quad h_y = \Xi(g_y) \\ &z = \Psi_E(h_x, h_y), \quad z_0 = \Psi_E(h_x, h_x) \\ &p(\hat{y}_t|y_{0:t-1}, x, z) = \Phi_D(y_{0:t-1}, g_x + \Psi_D(z) - \Psi_D(z_0)). \end{aligned} \tag{9}$$

The subtraction of $\Psi_D(z_0)$ in equation (9) prevents an over-representation of x as $\Psi_D(z)$ contains a coarse representation (aggregate form) of both sequences x and y. By modeling $p(z|x, y)$ our intent during training is to leverage coarse-grained $\Psi_D(z)$ and fine-grained g_x features representations, to capture the similarity between x and y.

2.1.3. Training and inference
We trained the model represented by equation (9), by minimizing the following loss function

$$L_{\text{WEISS}}(x, y) = L_{\text{NLL}}(x, y) + \lambda\left(\text{MMD}(p, q) + \|z_0\|^2\right), \tag{10}$$

where $L_{\text{NLL}}(x, y)$ is the same NLL term as in equation (2), $p = p(z|x, y)$, $q = q(z)$, $\text{MMD}(p, q)$ is the term enforcing z to be a Gaussian distribution, $\|z_0\|^2$ enforces $\mathbb{E}[z|h_x, h_x] = z_0 = \Psi_E(h_x, h_x)$, and λ is a regularization term.

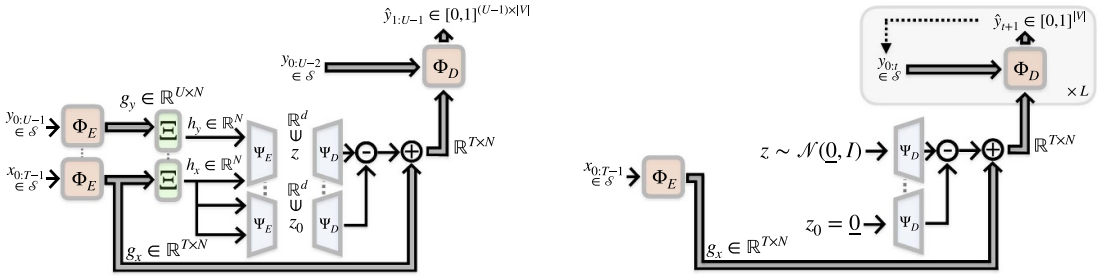

Figure 1. To the left, the training process of WEISS. Source and target, $x, y \in \mathcal{S}$, are fed into the same Φ_E (result in g_x and g_y), and the Ξ performs an average over the sequence dimension, producing $h_x, h_y \in \mathbb{R}^N$, associated to x and y, respectively. h_x and h_y are fed into Ψ_E, obtaining z and z_0, that are decoded to their original dimensionality N through Ψ_D. The vector $\Psi_D(z) - \Psi_D(z_0)$ is added to g_x before applying the conditioning to Φ_D. The right figure illustrates that at inference time we sample z from $\mathcal{N}(\underline{0}, I)$ and imposing $z_0 = \underline{0}$. At inference, the decoding step is repeated L times to generate a sequence.

At inference time z sampled from $\mathcal{N}(\underline{0}, I)$ and $z_0 = \underline{0}$ are firstly decoded by Ψ_D and then combined to the $\Phi_E(x)$, that is $g_x + \Psi_D(z) - \Psi_D(z_0)$, where x is the input sequence. Finally, this combined representation is passed to Φ_D along with the previous $t-1$ output tokens to generate the t token of the output sequence. Note that Ψ_E is discarded during inference. Figure 1 shows training (a) and inference (b) flows for WEISS.

3. Related works

In this work, we position WEISS within the context of efficient sampling strategies for autoregressive models. We categorize existing research into two primary approaches: (1) modeling latent variables within the encoder–decoder architecture and (2) enhancing decoding strategies.

3.1. Modeling latent variables

A significant number of studies investigated the potential of VAE based methods in conjunction with RNN (recurrent neural network) and LSTM to increase diversity by using the final hidden state of the encoder [23] or pooling it [24].

Cao *et al* [25] further enhanced the generation diversity of RNNs by employing both source and target encodings during training to model a latent variable, which are combined with the RNN state of the source to generate output sequences. A key difference to our approach is that the bias contribution introduced by the latent variable is not taken into account. We show in our experiments that this leads to deteriorate the correlation between inputs and outputs. Park *et al* [26] used a similar VAE-based strategy as [25], but applied to transformer models. In this method, the source sequence is pooled into a single vector representing the latent variable and used to condition the decoder. Unlike in WEISS, the target sequence was not considered in the construction of the latent variable, again leading to the loss of input–output correlation. Zhang *et al* [27] proposed a Wasserstein regularization for autoregressive models for time-series forecast. Here, the Wasserstein regularization is applied to one-dimensional spaces, allowing for an analytical solution. WEISS also applies a Wasserstein regularization but to higher dimensional spaces for improving the correlations between pairs of inputs, such as molecules, rather than individual inputs.

Relatively few studies have modeled latent variables to address the generation of novel molecules in de novo drug design. Zhou *et al* [28] proposed a variant that reduces the variance of the latent variable via multi-stage VAE to improve the properties of the generated molecules. Another approach proposed in [29] utilizes tied two-way transformers for both retrosynthesis and reaction prediction. In this method, a discrete latent variable is concatenated to the original sequences, to enhance the validity, diversity and reaction feasibility accuracy. However, the method proposed by [29] requires explicit marginalization on the latent variable, which becomes computationally expensive when the number of categories representing the latent variable increases. In contrast, WEISS samples the latent variable from a continuous distribution, thus avoiding explicit marginalization.

3.2. Enhancing decoding strategies

Many efficient decoding strategies have been proposed to enhance token diversity and address the issue of repetitive text from greedy search [30]: multinomial, top-k [31], top-p [16], unique randomizer [32], arithmetic sampling [33], and mask-predict [34], use the temperature parameter to control the level of diversity by adjusting the degree of randomness. Higher temperatures flatten probability distribution, giving

less likely tokens a higher chance of being selected. This reduces the dominance of the most likely tokens, leading to more diverse and varied output. Advanced tree-based strategies like priority sampling [35] yield high-quality, diverse samples but require higher computational costs due to tree branching at each step. Recent research shows larger models outperform smaller ones [36] but with higher computational costs. Speculative decoding [37], further improved by Kim *et al* [38] and Hu *et al* [39], it aims to mitigate this by using a smaller model to sample tokens under a larger model's guidance, while lookahead decoding [40] improves efficiency through parallel token generation.

Other strategies to improve sample efficiency involve float quantization [41], parameters pruning [42] and [43], deeper encoders and smaller decoders [44], multi-token prediction [45], and diversity with evolutionary algorithms [46]. All these approaches are not complementary to modelling latent variable strategies, and as such they can be seamlessly applied together with WEISS. In this paper, we focused on simple multinomial sampling or greedy search that proved the most effective in our experiments but other combinations are possible and left for future works.

4. Experiments on drug discovery tasks

To demonstrate the effectiveness of WEISS in preventing mode collapse while maintaining crucial features from the conditioned input, we carry out experiments on common tasks in the pharmaceutical domain. First, we show that a prior model with WEISS can be successfully guided by RL to generate novel molecules with enhanced properties (molecular optimization). Second, we show that WEISS applied in single-step retrosynthesis prediction yields diverse and feasible reactions.

4.1. Molecular optimization

Molecular optimization is the process of introducing small modifications to a source molecule in order to create a target molecule with improved properties. We considered the molecular pairs dataset used by [4] consisting of 6543 684, 418 180, and 475 070 unique pairs, for training, validation, and test, respectively. We compared the original Mol2Mol [4] model with Mol2Mol + WEISS, the extension incorporating our proposed framework. Moreover, we evaluated Mol2Mol + WEISS-B, the architecture in equation (5) trained without the bias correction term $\|z_0\|^2$ in the loss function in equation (2). Mol2Mol + WEISS-B can be seen as the transformer adaptation of the work by [25], and allows us to evaluate the impact of the bias correction term. We also evaluated Mol2Mol + VAE, a compatible implementation of the [26] method. In this approach, the decoder is conditioned by a single vector representing the source sequence. All models use multinomial sampling to generate tokens. The models are evaluated in terms of fraction of valid molecules (V), fraction of novel molecules (N), fraction of unique molecules (Uq), and similarity to the source molecule (S). Molecule validity is assessed by RDKit. Because different SMILES can correspond to the same molecule, the SMILES are first canonicalized using RDKit before computing molecular uniqueness. Details on training, model hyperparameters, and model selection are reported in the supplementary material.

4.1.1. Efficiency of WEISS

Table 1 reports a comparison of throughput (tokens produced per second) using Mol2Mol with beam search and Mol2Mol + WEISS. The throughput remains stable for Mol2Mol + WEISS, while it significantly drops for beam search. Our Mol2Mol + WEISS thus improves token throughput, consistently reducing inference costs when increasing the number of predictions.

4.1.2. Effect of varying the sampler

Figure 2 compares Mol2Mol with and Mol2Mol + WEISS using multinomial sampling at various temperatures $\tau \in \{0.1, 0.5, 1.0\}$. The NLLs consistently remain lower for Mol2Mol + WEISS than for Mol2Mol, especially for higher values of temperatures τ (see figure 2(a)). The fraction of unique sampled molecules is remarkably higher for Mol2Mol + WEISS than for Mol2Mol. Figure 2(b) shows that Mol2Mol + WEISS generates molecules with consistent average similarity of approximately 0.6 relative to the source compounds.

On the other hand, when the temperature τ decreases, the samples generated by Mol2Mol progressively become less diverse (Tanimoto similarity closer to 1) because of mode collapse.

4.1.3. Performance in molecular optimization task

Table 2 reports validity (V), novelty (N), uniqueness (Uq), and the fraction of generated molecules with Tanimoto similarity $\geqslant$0.5 to the source molecule ($S \geqslant 0.5$). We observed that Mol2Mol generates only 57% unique molecules because of mode collapse. In contrast, methods utilizing latent variables achieve between 73% and 99% uniqueness, indicating their effectiveness in reducing mode collapse.

Table 1. Throughput in tokens/sec for Mol2Mol2 (beam search) and Mol2Mol+WEISS (multinomial sampling, $d = 4$, $\lambda = 40$). B represents the number of sampled molecules. Higher throughputs denote better performance.

Model	$B = 10$	$B = 100$	$B = 1,000$
Mol2Mol	10 103 ± 2501	3319 ± 336	293 ± 25
WEISS	6957 ± 1550	6516 ± 1816	7245 ± 1700

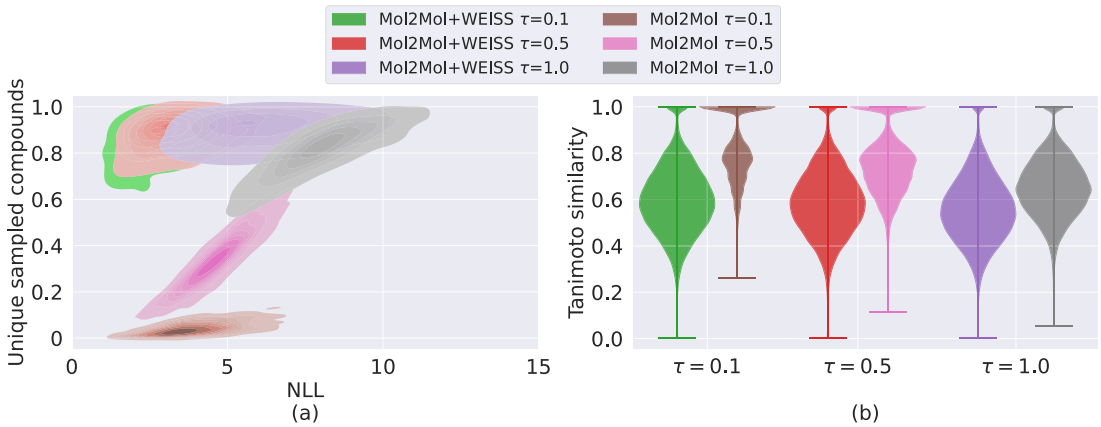

Figure 2. To the left (a), a comparison between Mol2Mol and Mol2Mol+WEISS. The x-axis shows the NLL, while the y-axis the percentage of unique and valid sampled compounds. The distributions are estimated by generating 1000 molecules for each test set source compounds using multinomial sampling at temperatures $\tau = 0.1$, $\tau = 0.5$, and $\tau = 1.0$. Mol2Mol+WEISS significantly produces a higher number of unique compounds maintaining low NLLs. Contrarily, Mol2Mol exhibits a different behaviour in function of τ: for $\tau = 0.1$ Mol2Mol mode collapses tending to generate always the compounds with high probability, while for $\tau = 1.0$ the number of diverse molecules increases along with the NLLs. To the right (b), the Tanimoto similarity distributions of the sampled molecules relative to their source compounds. Although the distributions for Mol2Mol+WEISS have lower averages compared to their Mol2Mol counterparts, the mode collapse is strongly mitigated, resulting in more varieties of sampled molecules as emerged from the left (a) illustration.

Table 2. Results for 1000 sampled molecules with multinomial sampling and temperature $\tau = 1.0$ from each test set source for Mol2Mol, Mol2Mol + VAE, Mol2Mol + WEISS-B, and Mol2Mol + WEISS with λ in parenthesis. For each model, we report the bottleneck dimensionality (d), the validity (V), the novelty (N), the uniqueness (Uq), and the percentage of molecules whose Tanimoto similarity to their corresponding sources is $\geqslant$0.5 ($S \geqslant 0.5$). For each entry, we report mean and standard deviation.

Model	d	V	N	Uq	$S \geqslant 0.5$
Mol2Mol	—	$0.98_{\pm 0.06}$	$0.99_{\pm 0.02}$	$0.57_{\pm 0.12}$	$0.87_{\pm 0.10}$
VAE	2	$0.97_{\pm 0.01}$	$0.91_{\pm 0.01}$	$0.99_{\pm 0.00}$	$0.00_{\pm 0.00}$
VAE	4	$0.99_{\pm 0.00}$	$0.79_{\pm 0.01}$	$0.99_{\pm 0.00}$	$0.00_{\pm 0.00}$
VAE	16	$0.98_{\pm 0.00}$	$0.85_{\pm 0.01}$	$0.99_{\pm 0.00}$	$0.00_{\pm 0.00}$
WEISS-B (5)	2	$0.91_{\pm 0.09}$	$0.99_{\pm 0.01}$	$0.92_{\pm 0.04}$	$0.34_{\pm 0.14}$
WEISS-B (10)	4	$0.87_{\pm 0.09}$	$0.99_{\pm 0.01}$	$0.99_{\pm 0.01}$	$0.26_{\pm 0.11}$
WEISS-B (40)	16	$0.91_{\pm 0.09}$	$0.99_{\pm 0.01}$	$0.97_{\pm 0.02}$	$0.28_{\pm 0.13}$
WEISS (5)	2	$0.98_{\pm 0.05}$	$0.99_{\pm 0.01}$	$0.73_{\pm 0.09}$	$0.75_{\pm 0.10}$
WEISS (10)	4	$0.92_{\pm 0.07}$	$0.99_{\pm 0.01}$	$0.86_{\pm 0.09}$	$0.60_{\pm 0.10}$
WEISS (40)	16	$0.94_{\pm 0.09}$	$0.99_{\pm 0.01}$	$0.90_{\pm 0.08}$	$0.56_{\pm 0.13}$

Mol2Mol + VAE, Mol2Mol + WEISS-B and Mol2Mol + WEISS all generate a large set of diverse molecules, but Mol2Mol + WEISS is the only method that better preserves the similarity ($\geqslant$56%) of the generated molecules to the source molecules (0% for Mol2Mol + VAE, and $\leqslant$ 34% for Mol2Mol + WEISS-B).

To complement table 2, in table 3 we extended our results to temperature $\tau \in \{0.1, 0.5\}$ confirming the same trends. In particular, we observed that WEISS well trades off between the two objectives, uniqueness and percentage of molecules with an $S \geqslant 0.5$ score, for both temperatures.

4.1.4. Comparison with beam search

As beam search is slower than multinomial sampling, we compared the results of Mol2Mol + Beam search with Mol2Mol + WEISS and multinomial sampling ($\tau = 1.0$). Table 4 shows that Mol2Mol + WEISS is able

Table 3. Results for 1000 sampled molecules that extend table 2 in the main paper. The table reports the validity (V), the novelty (N), the uniqueness (Uq), the percentage of molecules whose Tanimoto similarity to their corresponding sources is $\geqslant$0.5 ($S \geqslant 0.5$), and bottleneck dimensionality (d) of the model. We report the results also with $\tau \in \{0.1, 0.5\}$. Best results are highlighted in bold.

		$\tau = 0.1$				$\tau = 0.5$			
Model	d	V	N	Uq	$S \geqslant 0.5$	V	N	Uq	$S \geqslant 0.5$
Mol2Mol	—	$0.99_{\pm 0.07}$	$0.95_{\pm 0.015}$	$0.01_{\pm 0.02}$	$0.99_{\pm 0.10}$	$0.99_{\pm 0.06}$	$0.99_{\pm 0.04}$	$0.13_{\pm 0.05}$	$0.97_{\pm 0.10}$
VAE	2	$1.00_{\pm 0.00}$	$0.69_{\pm 0.04}$	$0.04_{\pm 0.00}$	$0.00_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.77_{\pm 0.01}$	$0.57_{\pm 0.02}$	$0.00_{\pm 0.00}$
VAE	4	$1.00_{\pm 0.00}$	$0.49_{\pm 0.05}$	$0.04_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.57_{\pm 0.02}$	$0.87_{\pm 0.01}$	$0.00_{\pm 0.00}$
VAE	16	$1.00_{\pm 0.00}$	$0.73_{\pm 0.02}$	$0.26_{\pm 0.01}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.67_{\pm 0.01}$	$0.93_{\pm 0.01}$	$0.00_{\pm 0.00}$
WEISS-B (5)	2	$0.94_{\pm 0.07}$	$1.00_{\pm 0.01}$	$0.49_{\pm 0.05}$	$0.60_{\pm 0.16}$	$0.92_{\pm 0.07}$	$1.00_{\pm 0.00}$	$0.72_{\pm 0.07}$	$0.57_{\pm 0.17}$
WEISS-B (10)	4	$0.88_{\pm 0.09}$	$1.00_{\pm 0.00}$	$0.95_{\pm 0.02}$	$0.35_{\pm 0.14}$	$0.88_{\pm 0.06}$	$1.00_{\pm 0.00}$	$0.96_{\pm 0.02}$	$0.32_{\pm 0.12}$
WEISS-B (40)	16	$0.92_{\pm 0.08}$	$1.00_{\pm 0.00}$	$0.89_{\pm 0.04}$	$0.39_{\pm 0.16}$	$0.93_{\pm 0.08}$	$1.00_{\pm 0.00}$	$0.93_{\pm 0.04}$	$0.37_{\pm 0.16}$
WEISS (5)	2	$0.99_{\pm 0.05}$	$0.99_{\pm 0.02}$	$0.35_{\pm 0.05}$	$\mathbf{0.87}_{\pm 0.10}$	$0.99_{\pm 0.05}$	$0.99_{\pm 0.01}$	$0.49_{\pm 0.08}$	$\mathbf{0.84}_{\pm 0.10}$
WEISS (10)	4	$\mathbf{0.95}_{\pm 0.07}$	$\mathbf{1.00}_{\pm 0.01}$	$\mathbf{0.71}_{\pm 0.09}$	$\mathbf{0.72}_{\pm 0.10}$	$\mathbf{0.95}_{\pm 0.06}$	$\mathbf{1.00}_{\pm 0.01}$	$\mathbf{0.76}_{\pm 0.09}$	$\mathbf{0.70}_{\pm 0.10}$
WEISS (40)	16	$\mathbf{0.97}_{\pm 0.07}$	$\mathbf{1.00}_{\pm 0.01}$	$\mathbf{0.76}_{\pm 0.10}$	$0.71_{\pm 0.14}$	$\mathbf{0.96}_{\pm 0.07}$	$\mathbf{1.00}_{\pm 0.01}$	$\mathbf{0.81}_{\pm 0.10}$	$0.68_{\pm 0.14}$

Table 4. Results for 1000 sampled molecules with Mol2Mol + WEISS (λ) with multinomial sampling and temperature $\tau = 1.0$ and Mol2Mol with beam search from a subset of 1000 test set source molecules. For each model, we report the bottleneck dimensionality (d), the validity (V), the novelty (N), the uniqueness (Uq), the percentage of molecules whose Tanimoto similarity to their corresponding sources is $\geqslant$0.5 ($S \geqslant 0.5$), and total execution time. For each entry (with the exception of time), we report mean and standard deviation.

Model	d	V	N	Uq	$S \geqslant 0.5$	Time
Mol2Mol +WEISS (5)	2	$0.98_{\pm 0.06}$	$1.00_{\pm 0.01}$	$0.73_{\pm 0.09}$	$0.75_{\pm 0.11}$	5 h 46 min
Mol2Mol + WEISS (10)	4	$0.92_{\pm 0.07}$	$1.00_{\pm 0.01}$	$0.86_{\pm 0.09}$	$0.60_{\pm 0.10}$	5 h 48 min
Mol2Mol + WEISS (40)	16	$0.95_{\pm 0.08}$	$0.99_{\pm 0.01}$	$0.90_{\pm 0.08}$	$0.58_{\pm 0.13}$	5 h 55 min
Mol2Mol + Beam search	—	$0.98_{\pm 0.06}$	$0.99_{\pm 0.01}$	$0.97_{\pm 0.03}$	$0.88_{\pm 0.12}$	70 h 1 min

to generate a high fraction of unique molecules with a high percentage of similar molecules while being 14 times faster than Mol2Mol + Beam search. These results confirm we can preserve the right tradeoff between V, N and Uq with less computation. Moreover, as the beam size (number of sampled molecules) rises the more the time gap increases between Mol2Mol + Beam search and Mol2Mol + WEISS.

4.1.5. Molecular properties optimization with RL

To further demonstrate the explorative capabilities of the model, we coupled our framework with RL to generate enhanced compounds, starting from the same four compounds used by [9] and depicted in figure 3. Those compounds were chosen from ExCAPE-DB [47] due to their potency (pIC50 > 5) and activity against the dopamine receptor D2 (DRD2). With RL, we aim to generate molecules that maximize the following scores:

- $\text{active}(y) \in [0, 1]$: the probability of a structure being active against DRD2 [48], predicted by a random forest model.
- $\text{qed}(y) \in [0, 1]$: the quantitative estimate of drug-likeness (QED) computed by RDKit. A higher score indicates that the molecule has more drug-like properties.
- $\text{sim}(x, y) \in [0, 1]$: Tanimoto similarity to the source molecule x.

To jointly maximizes the three scores, we aggregated them by using the geometric mean

$$s(y) = \sqrt[3]{\text{active}(y) \cdot \text{qed}(y) \cdot \text{sim}(x, y)}. \tag{11}$$

Mol2Mol and Mol2Mol + WEISS ($d = 4$) were fine-tuned with REINVENT4 [8] as RL framework. We conducted multiple RL fine-tuning sessions, varying the multinomial sampling temperature τ. More training details are reported in the supplementary material. Figure 3 shows that Mol2Mol + WEISS consistently generated more molecules with improved scores than Mol2Mol. Note that the molecules at the extremes are difficult to optimize, as their QED or activity values are close to the maximum. Even so, Mol2Mol + WEISS appears successful in this task.

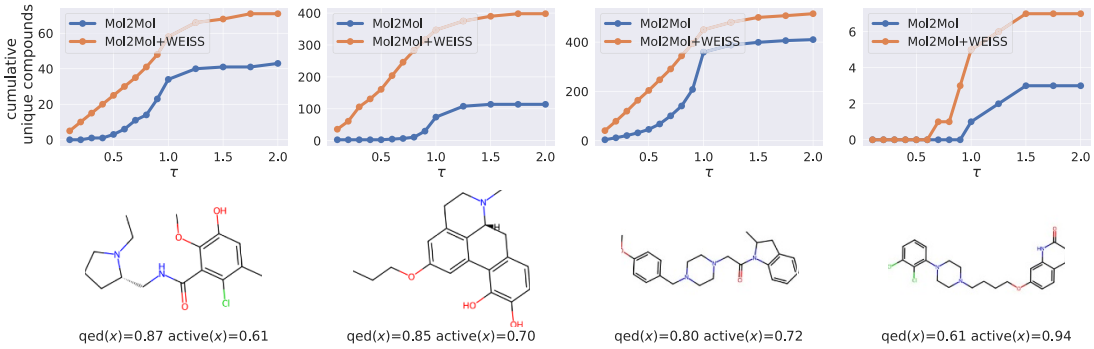

Figure 3. The cumulative sum of molecules with higher scores $s(y)$ than the source x molecule is reported on the y-axis. The x-axis represents different temperatures τ used for generating compounds with multinomial sampling. Source molecules are displayed below each plot. Mol2Mol + WEISS consistently generated more molecules than Mol2Mol, even for molecules that are difficult to optimize (left and right plots).

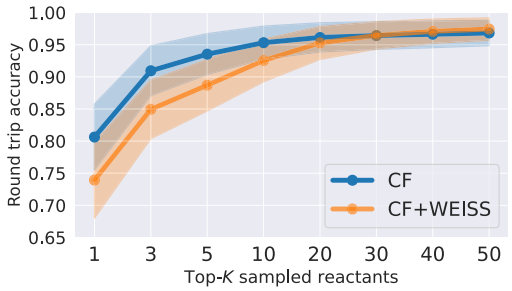

Figure 4. Comparison of top-K round-trip accuracy between Chemformer with beam search (CF) and Chemformer with WEISS and greedy search (CF + WEISS).

4.2. Single-step retrosynthesis prediction

While generating novel molecules with specific properties is a major challenge in drug discovery, another challenge resides in designing the process for making these molecules [49] i.e. the synthesis planning [50]. In single-step retrosynthesis, a model, e.g. a transformer, is used to predict building block molecules (reactants) that should react together to form the given input target molecule (product) [51]. This step can be applied iteratively [52] until all building blocks are available for purchase [53].

In general, the model should learn both basic and complex chemistry, including identification of reactive sites and prioritization of possible reaction rules. Notably, this problem is not trivial [54]. For example, atoms or groups of atoms can be 'lost' as byproducts during the reaction. Because different sets of reactants can form the same product molecule, it is advantageous with a sampling strategy that can produce diverse, yet feasible, sets of reactants.

4.2.1. Model training and evaluation metrics

We used the Chemformer [55], a BART-based language model pre-trained on SMILES for retrosynthesis prediction.

We compared two Chemformers: (1) standard with beam search (CF) and (2) WEISS-modified with greedy search (CF + WEISS). Both models were fine-tuned for 50 epochs on USPTO-Full, a public dataset with chemical reactions [56]. The dataset preparation was done using AiZynthTrain [57] and [6], splitting the dataset randomly into training, validation and test sets of 943.6 k, 52.4 k, and 52.4 k (90:5:5), respectively. CF + WEISS was fine-tuned using CF as initial state.

To evaluate the models, we employ top-K round-trip accuracy (RTA), a standard as a computational proxy for reaction feasibility in the retrosynthesis prediction field [58]. RTA is computed using a product-prediction Chemformer to validate retrosyntehsis predictions. We compute RTA by feeding the predicted reactants to the model and generating one product. A retrosynthesis prediction is considered correct if the predicted product corresponds to the input product molecule. RTA thus accounts for the fact that a single product can be synthesized in multiple ways. Because the product-prediction Chemformer is trained on historic reaction data, a high RTA indicates simplicity and feasibility of the predicted reactions,

Table 5. Fraction of valid molecules generated (V), and the proportion of valid and unique reactants (Uq) using Chemformer with beam search (CF) and WEISS with greedy search (CF + WEISS), where 50 random latent variables are sampled with WEISS. Best results are highlighted in bold.

Model	V	Uq
CF	**1.00** ± 0.00	0.34 ± 0.03
CF + WEISS ($\lambda = 20$, $d = 4$)	**1.00** ± 0.00	**0.52** ± 0.04

while novel reactions are more difficult to validate [6]. In addition to RTA, we report the fraction of valid molecules (V) and the fraction of unique sets of reactant molecules (Uq). We report mean and standard deviation over inference batches.

4.2.2. RTA and diversity of predictions

Figure 4 shows the top-K RTA. The fraction of valid and uniquely sampled reactants molecules are reported in table 5. Both models maintain high validity of the sampled SMILES. Interestingly, the percentage of unique molecules is significantly higher for CF + WEISS, while the RTA is close to that of CF. Hence, CF + WEISS enables generating feasible reactions on par with beam search while exploring more of the chemical space.

5. Conclusion

In this paper, we presented WEISS, a novel framework for training and efficient sampling, that enhances the output diversity while preserving its integrity. Our WEISS framework leverages a continuous latent variable to expand the expressiveness of the higher-dimensional representations of each token. The experiments on drug discovery tasks demonstrate that the proposed mechanism not only increases the diversity but also maintains high levels of similarity with the conditioning input. Furthermore, WEISS is especially advantageous compared to beam search in applications where a large number of predictions are generated iteratively, as in case of molecular optimization with RL. Notably, our model effectively balances the trade-off between generating diverse outputs and ensuring that these outputs remain closely correlated with the input. Experiments on drug design suggest that WEISS is a versatile framework, applicable to a broader category of autoregressive encoder–decoder models.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://github.com/r1cc4r2o/weiss [18].

ORCID iDs

Riccardo Tedoldi https://orcid.org/0009-0002-1477-0121
Ola Engkvist https://orcid.org/0000-0003-4970-6461
Andrea Passerini https://orcid.org/0000-0002-2765-5395
Annie M Westerlund https://orcid.org/0000-0003-2288-5711
Alessandro Tibo https://orcid.org/0000-0002-9070-740X

References

[1] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser L and Polosukhin I 2017 Attention is all you need *Advances in Neural Information Processing Systems* vol 30 (Curran Associates, Inc.) (available at: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf)
[2] Brown T *et al* 2020 Language models are few-shot learners *Advances in Neural Information Processing Systems* vol 33 (Curran Associates, Inc.) pp 1877–901 (available at: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf)
[3] Arús-Pous J, Blaschke T, Ulander S, Reymond J-L, Chen H and Engkvist O 2019 Exploring the GDB-13 chemical space using deep generative models *J. Cheminform.* **11** 20
[4] He J, Nittinger E, Tyrchan C, Czechtizky W, Patronov A, Bjerrum E J and Engkvist O 2022 Transformer-based molecular optimization beyond matched molecular pairs *J. Cheminform.* **14** 18
[5] Schwaller P, Laino T, Gaudin T, Bolgar P, Hunter C A, Bekas C and Lee A A 2019 Molecular transformer: a model for uncertainty-calibrated chemical reaction prediction *ACS Cent. Sci.* **5** 1572–83
[6] Westerlund A M, Manohar Koki S, Kancharla S, Tibo A, Saigiridharan L, Kabeshov M, Mercado R and Genheden S 2024 Do chemformers dream of organic matter? Evaluating a transformer model for multistep retrosynthesis *J. Chem. Inf. Model.* **64** 3021–33

[7] Chen Y, Wang Z, Wang L, Wang J, Li P, Cao D, Zeng X, Ye X and Sakurai T 2023 Deep generative model for drug design from protein target sequence *J. Cheminform.* **15** 38

[8] Loeffler H H, He J, Tibo A, Janet J P, Voronov A, Mervin L H and Engkvist O 2024 Reinvent 4: modern AI–driven generative molecule design *J. Cheminform.* **16** 20

[9] He J, Tibo A, Janet J P, Nittinger E, Tyrchan C, Czechtizky W and Engkvist O 2024 Evaluation of reinforcement learning in transformer-based molecular design *J. Cheminform.* **16** 95

[10] Pajouhesh H and Lenz G R 2005 Medicinal chemical properties of successful central nervous system drugs *NeuroRx* **2** 541–53

[11] Vieth M, Siegel M G, Higgs R E, Watson I A, Robertson D H, Savin K A, Durst G L and Hipskind P A 2004 Characteristic physical properties and structural fragments of marketed oral drugs *J. Med. Chem.* **47** 224–32

[12] Sutskever I, Vinyals O and Le Q V 2014 Sequence to sequence learning with neural networks *Proc. 27th Int. Conf. on Neural Information Processing Systems NIPS'14* vol 2 (MIT Press) pp 3104–12 (available at: https://proceedings.neurips.cc/paper_files/paper/2014/file/5a18e133cbf9f257297f410bb7eca942-Paper.pdf)

[13] Weininger D 1988 SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules *J. Chem. Inf. Comput. Sci.* **28** 31–36

[14] Ahn S, Chen B, Wang T and Song L 2022 Spanning tree-based graph generation for molecules *Int. Conf. on Learning Representations* (available at: https://openreview.net/pdf?id=w60btE_8T2m)

[15] Jin W, Barzilay R and Jaakkola T 2018 Junction tree variational autoencoder for molecular graph generation *Proc. 35th Int. Conf. on Machine Learning, Volume 80 of Proc. of Machine Learning Research* ed J Dy and A Krause (PMLR) pp 2323–32 (available at: https://proceedings.mlr.press/v80/jin18a/jin18a.pdf)

[16] Holtzman A, Buys J, Du Li, Forbes M and Choi Y 2020 The curious case of neural text degeneration *8th Int. Conf. on Learning Representations, ICLR 2020 (Addis Ababa, Ethiopia)* (available at: https://openreview.net/pdf?id=rygGQyrFvH)

[17] Tolstikhin I, Bousquet O, Gelly S and Schoelkopf B 2018 Wasserstein auto-encoders *Int. Conf. on Learning Representations* (available at: https://openreview.net/pdf?id=HkL7n1-0b)

[18] Tedoldi R, Li J, Engkvist O, Passerini A, Westerlund A and Tibo A 2025 WEISS: Wasserstein efficient sampling strategy for LLMs in drug design GitHub repository, r1cc4r2o/weiss (available at: https://github.com/r1cc4r2o/weiss)

[19] Hochreiter S and Schmidhuber J 1997 Long short-term memory *Neural Comput.* **9** 1735–80

[20] Raffel C, Shazeer N M, Roberts A, Lee K, Narang S, Matena M, Zhou Y, Li W and Liu P J 2019 Exploring the limits of transfer learning with a unified text-to-text transformer *J. Mach. Learn. Res.* **21** 1–140:67 (available at: https://jmlr.org/papers/volume21/20-074/20-074.pdf)

[21] Tibo A, He J, Janet J P, Nittinger E and Engkvist O 2024 Exhaustive local chemical space exploration using a transformer model *Nat. Commun.* **15** 7315

[22] Kingma D P and Welling M 2014 Auto-encoding variational bayes *2nd Int. Conf. on Learning Representations, ICLR 2014, (Banff, AB, Canada, 14–16 April 2014), (Conf. Track Proc.)* ed Y Bengio and Y LeCun (available at: https://openreview.net/pdf?id=33X9fd2-9FyZd)

[23] Bowman S R, Vilnis L, Vinyals O, Dai A, Jozefowicz R and Bengio S 2016 Generating sentences from a continuous space *Proc. 20th SIGNLL Conf. on Computational Natural Language Learning* ed S Riezler and Y Goldberg (Association for Computational Linguistics) pp 10–21

[24] Zhang B *et al* 2016 *Proc. 2016 Conf. on Empirical Methods in Natural Language Processing (Austin, Texas)* (Association for Computational Linguistics) pp 521–30

[25] Cao K and Clark S 2017 Latent variable dialogue models and their diversity *Proc. 15th Conf. of the European Chapter of the Association for Computational Linguistics (Short Papers)* vol 2, ed M Lapata, P Blunsom and A Koller (Association for Computational Linguistics) pp 182–7 (available at: https://aclanthology.org/E17-2029.pdf)

[26] Park S and Lee J 2021 Finetuning pretrained transformers into variational autoencoders *Proc. 2nd Workshop on Insights From Negative Results in NLP (Online and Punta Cana, Dominican Republic)* ed J ao Sedoc, A Rogers, A Rumshisky and S Tafreshi (Association for Computational Linguistics)pp 29–35

[27] Zhang C, Kokoszka P and Petersen A 2020 Wasserstein autoregressive models for density time series (arXiv:2006.12640)

[28] Zhou C and Poczos B 2023 Objective-agnostic enhancement of molecule properties via multi-stage vae (arXiv:2308.13066)

[29] Kim E, Lee D, Kwon Y, Sik Park M and Choi Y-S 2021 Valid, plausible and diverse retrosynthesis using tied two-way transformers with latent variables *J. Chem. Inf. Model.* **61** 123–33

[30] Schuurmans D 1999 Greedy importance sampling *Advances in Neural Information Processing Systems* vol 12, ed S Solla, T Leen and K Müller (MIT Press) (available at: https://proceedings.neurips.cc/paper_files/paper/1999/file/8303a79b1e19a194f1875981be5bdb6f-Paper.pdf)

[31] Shi S, Chu X, Chun Cheung K and See S 2019 Understanding top-k sparsification in distributed deep learning (arXiv:1911.08772)

[32] Shi K, Bieber D and Sutton C 2020 Incremental sampling without replacement for sequence models *Proc. 37th Int. Conf. on Machine Learning, Volume 119 of Proc. Machine Learning Research* ed H Daumé and A Singh (PMLR) pp 8785–95 (available at: http://proceedings.mlr.press/v119/shi20a/shi20a.pdf)

[33] Vilnis L, Zemlyanskiy Y, Murray P, Passos A and Sanghai S 2023 Arithmetic sampling: parallel diverse decoding for large language models *Proc. 40th Int. Conf. on Machine Learning* p ICML'23 (available at: https://openreview.net/pdf?id=EfhmBBrXY2)

[34] Ghazvininejad M, Levy O, Liu Y and Zettlemoyer L 2019 Mask-predict: parallel decoding of conditional masked language models *Proc. 2019 Conf. on Empirical Methods in Natural Language Processing and the 9th Int. Joint Conf. on Natural Language Processing (EMNLP-IJCNLP) (Hong Kong, China)* ed K Inui, J Jiang, V Ng and X Wan (Association for Computational Linguistics) pp 6112–21

[35] Grubisic D, Cummins C, Seeker V and Leather H 2024 Priority sampling of large language models for compilers (arXiv:2402.18734)

[36] Kaplan J, McCandlish S, Henighan T, Brown T B, Chess B, Child R, Gray S, Radford A, Wu J and Amodei D 2020 Scaling laws for neural language models (arXiv:2001.08361)

[37] Leviathan Y, Kalman M and Matias Y 2023 Fast inference from transformers via speculative decoding *Proc. 40th Int. Conf. on Machine Learning* ICML'23 (available at: https://openreview.net/pdf?id=C9NEblP8vS)

[38] Kim S, Mangalam K, Moon S, Malik J, Mahoney M W, Gholami A and Keutzer K 2024 Speculative decoding with big little decoder *Proc. 37th Int. Conf. on Neural Information Processing Systems, NIPS'23,* (Curran Associates Inc.) (available at: https://openreview.net/pdf?id=EfMyf9MC3t)

[39] Hu Z and Huang H 2024 Accelerated speculative sampling based on tree monte carlo *41st Int. Conf. on Machine Learning* (available at: https://openreview.net/pdf?id=stMhi1Sn2G)

[40] Fu Y, Bailis P, Stoica I and Zhang H 2024 Break the sequential dependency of llm inference using lookahead decoding (arXiv:2402.02057)

[41] Dettmers T, Lewis M, Belkada Y and Zettlemoyer L 2022 GPT3.int8: 8-bit matrix multiplication for transformers at scale *Advances in Neural Information Processing Systems* ed A H Oh, A Agarwal, D Belgrave and K Cho (available at: https://proceedings.neurips.cc/paper_files/paper/2022/file/c3ba4962c05c49636d4c6206a97e9c8a-Paper-Conference.pdf)
[42] Fan A, Grave E and Joulin A 2020 Reducing transformer depth on demand with structured dropout *Int. Conf. on Learning Representations* (available at: https://openreview.net/pdf?id=SylO2yStDr)
[43] Kwon W, Kim S, Mahoney M W, Hassoun J, Keutzer K and Gholami A 2022 A fast post-training pruning framework for transformers *Advances in Neural Information Processing Systems NIPS'22* vol 35, ed S Koyejo, S Mohamed, A Agarwal, D Belgrave, K Cho and A Oh (Curran Associates, Inc.) pp 24101–16 (available at: https://proceedings.neurips.cc/paper_files/paper/2022/file/987bed997ab668f91c822a09bce3ea12-Paper-Conference.pdf)
[44] Kasai J, Pappas N, Peng H, Cross J and Smith N 2021 Deep encoder, shallow decoder: Reevaluating non-autoregressive machine translation *Int. Conf. on Learning Representations* (available at: https://openreview.net/forum?id=KpfasTaLUpq)
[45] Gloeckle F, Youbi Idrissi B, Roziére B, Lopez-Paz D and Synnaeve G 2024 Better & faster large language models via multi-token prediction (arXiv:2404.19737)
[46] So D R, Liang C and Le Q V 2019 The evolved transformer *Proc. 36th Int. Conf. on Machine Learning (Proc. of Machine Learning Research ICML'19* vol 97), ed K Chaudhuri and R Salakhutdinov (PMLR) pp 5877–86 (available at: https://proceedings.mlr.press/v97/so19a/so19a.pdf)
[47] Sun J *et al* 2017 ExCAPE-DB: an integrated large scale dataset facilitating big data analysis in chemogenomics *J. Cheminform.* **9** 1–9
[48] Olivecrona M, Blaschke T, Engkvist O and Chen H 2017 Molecular de-novo design through deep reinforcement learning *J. Cheminform.* **9** 48
[49] Oliveira J C A, Frey J, Zhang S-Q, Xu L-C, Li X, Li S-W, Hong X and Ackermann L 2022 When machine learning meets molecular synthesis *Trends Chem.* **4** 863–85
[50] Robinson R 1917 LXIII–a synthesis of tropinone *J. Chem. Soc. Trans.* **111** 762–8
[51] Genheden S, Thakkar A, Chadimová V, Reymond J-L, Engkvist O and Bjerrum E J 2020 AiZynthFinder: a fast, robust and flexible open-source software for retrosynthetic planning *J. Cheminform.* **12** 70
[52] Thakkar A, Kogej T, Reymond J-L, Engkvist O and Jannik Bjerrum E 2020 Datasets and their influence on the development of computer assisted synthesis planning tools in the pharmaceutical domain *Chem. Sci.* **11** 154–68
[53] Segler M H S and Waller M P 2017 Neural-symbolic machine learning for retrosynthesis and reaction prediction *Chem. A Euro. J.* **23** 5966–71
[54] Segler M H S, Preuss M and Waller M P 2018 Planning chemical syntheses with deep neural networks and symbolic ai *Nature* **555** 604–10
[55] Irwin R, Dimitriadis S, He J and Bjerrum E J 2022 Chemformer: a pre-trained transformer for computational chemistry *Mach. Learn.: Sci. Technol.* **3** 015022
[56] Dai H, Li C, Coley C, Dai B and Song L 2019 Retrosynthesis prediction with conditional graph logic network *Advances in Neural Information Processing Systems NIPS'19* vol 32, ed H Wallach, H Larochelle, A Beygelzimer, F d Alché-Buc, E Fox and R Garnett (Curran Associates, Inc.) (available at: https://proceedings.neurips.cc/paper_files/paper/2019/file/0d2b2061826a5df3221116a5085a6052-Paper.pdf)
[57] Genheden S, Norrby P-O and Engkvist O 2023 AiZynthTrain: robust, reproducible and extensible pipelines for training synthesis prediction models *J. Chem. Inf. Model.* **63** 1841–6
[58] Schwaller P, Petraglia R, Zullo V, Nair V H, Haeuselmann R A, Pisoni R, Bekas C, Iuliano A and Laino T 2020 Predicting retrosynthetic pathways using transformer-based models and a hyper-graph exploration strategy *Chem. Sci.* **11** 3316–25